Passagiere und Besatzungsmitglieder in Fahrzeugen

Image not found or type unknown

Deutsch

Viele offene Fahrzeuge sehen besser aus, wenn sie Besatzungsmitglieder und Passagiere enthalten und nicht völlig leer in der Gegend herumtuckern.

Zombies zu verteilen ist gar nicht schwierig und in drei einfachen Schritten erledigt. Öffnet die .mdl des Modells und baut folgende Zeilen ein:

1. Ganz am Anfang, noch vor der Zeile "function data ()":

Code

```
local vec3 = require "vec3"
local transf = require "transf"
```

Diese beiden Zeilen erklären dem Programm, was es mit den Funktionen vec3 und transf anfangen soll. Fehlen die Zeilen, kommt der CTD mit Fehlermeldung.

2. Ganz am Ende der .mdl findet ihr den Block "transportVehicle = {" ...

Hinter die Schweifklammer, die den Unterblock "capacities=..." wieder schließt, fügt den Block...

Code

```
seats = {

}
```

...ein.

3. Zwischen Zeilen 1 und 5 könnt ihr dann beliebig viele Zombies aufrufen. Das geschieht mit folgendem Befehl, den ihr einfach für jeden gewünschten Zombie einmal einbaut:

Code

```
{ group = 17, transf = transf.rotZYXTransl(transf.degToRad(-10, -12, 0.0), vec3.new(9.8,1,-8
```

Für TPF2 kommen noch mehr Informationen dazu:

Code

```
{ animation = "sitting", crew = false, forward = true, group = 17, transf = transf.rotZYXTra
```

Die Zahlen, die in dieser Zeile stehen, sind folgende Werte:

{ group = p, transf = transf.rotZYXTransl(transf.degToRad(a, b, c), vec3.new(x, y, z), crew = v, standing = w},

p die laufende Nummer in der .mdl des Teils, an dem die Zombies "befestigt" werden sollen. Unter "children=" stehen die ganzen .mdls und .grps, aus denen das Modell besteht; zählt einfach von oben durch. Möglich ist, die Zombies an das Teil zu hängen, das auch ihre Sitze enthält, man kann sie aber auch an die Räder hängen und ewig Purzelbäume schlagen lassen oder wie auch immer man mag. Oft wird es die beste Option sein, die Zombies an Teil 0 zu hängen: das ist das Modell selbst; so spart man sich die Probleme, die unter Umständen entstehen, wenn das Teil in höheren LoDs ausgeblendet wird.

a,b,c: die Drehwinkel der Zombies um die zyx-Achsen; 0,0,0 ist senkrecht stehend, Blickrichtung in Fahrtrichtung frei geradeaus.

x,y,z: die Koordinaten, an dem der Zombie sich aufhalten soll. Vorsicht: 0,0,0 ist der Ursprung des Teils p, an dem der Zombie dranhängt, und nicht die absoluten Koordinaten! Das heißt also, daß ihr für die korrekten Koordinaten des Zombies die Koordinaten des Ursprungs von den gewünschten absoluten Koordinaten des Zombies abziehen müßt. Der Modelviewer ist hier sehr hilfreich; es braucht meist mehrere Versuche, bis sie genau da sitzen, wo man sie haben möchte.

v: Boolesche Variable; bei "true" gibt es diesen griesgrämigen Gesellen, dem die ~~Håckn~~ Arbeit auf gut Wienerisch am Oasch geht, und der immer da ist, sobald das Fahrzeug sich bewegt und nicht als Asset rumsteht. "False" bringt Passagiere; in dem Fall definiert die Zeile nur einen möglichen Passagiersitz und das

Spiel besetzt ihn je nach Passagierzahl, die im Fahrzeug/Zug gerade vorhanden ist, oder läßt ihn frei, wenn er gerade nicht gebraucht wird.

w: ebenfalls Boole. "True" stellt den Zombie hin, "False" setzt ihn auf den Hintern.

Für TPF2 kann man verschiedene Animationen festlegen:

"sitting" ist für Passagiere geeignet; sie sind allerdings relativ mobil.

"driving" paßt für Autos und andere Fahrzeuge, in denen man tief drinsitzt.

"driving_upright" ist für Busse, Züge oder Flugzeuge geeignet.

Für Besatzungen gehört in TPF2 auch noch eine weitere Information hinzu:

Code

```
drivingLicense = "x",
```

...definiert das Aussehen des Besatzungsmitglieds. Folgende Optionen gibt es hier:

"AIR": Pilot in grauer Uniform

"AIR_OUTDOOR": Pilot mit Fliegerbrille und Schal

"BUS": Busfahrer

"TRUCK": Lastwagenfahrer

"TRAM": Straßenbahnfahrer

"RAIL": Lokführer

"WATER": Kapitän.

Die obige Beispielzeile ruft also einen sitzenden Passagier auf, der 10° nach hinten lehnt und 12° um die Hochachse gedreht wurde sowie relativ zum Teil Nummer 17 knapp 10m vorne, 1m nach rechts und knapp 9m weiter unten positioniert ist.

Image not found or type unknown
:)

Die Anzahl der Passagiersitze ist dem Spiel grundsätzlich egal; sie werden bei fehlenden Sitzen proportional zur Anzahl der mitfahrenden Leute besetzt. Man kann also durchaus von 20 Passagiersitzen nur 10 auf dem Sonnendeck darstellen und die übrigen 10, die irgendwo im Kohlenbunker sitzen, gar nicht erst definieren.

Wenn man allerdings 10 Passagiersitze definiert und nur 5 Passagiere gemäß .mdl Platz finden, dann sitzen auch nur maximal 5 Zombies zufällig verteilt da und die fünf übrigen Plätze bleiben frei.

Eine Warnung: Zombies sitzen nicht still. Sie zappeln herum und lassen sich auch von massivem Blech nicht aufhalten. Es lohnt sich deswegen, einmal die Zombies im Modelviewer anzusehen. Wenn sie Körperteile durch die Wand rammen, dreht sie einfach 20° nach innen (in der obigen Zeile bei b) oder verschiebt sie ein wenig nach Bedarf.

Viel Spaß!

Image not found or type unknown

English

Many open vehicles look much better when they are not driving around empty. A few zombie passengers and crew members work wonders on their realism.

Summoning them is not hard at all and easily done using three simple curses that need to be added to the vehicles .mdl file.

First of all, you need to explain the game two functions that are required for the zombies placement.

Code

```
local vec3 = require "vec3"
local transf = require "transf"
```

These two lines need to be inserted right at the very beginning of the .mdl in front of everything else. If you try to call for the zombies without this preparation, the game will crash.

Secondly, look for the section "transportVehicle=..." that is typically located at or near the end of the file. In there, you will find a block "capacities={" that is closed with another curly bracket "}". Right behind this closing bracket, add another block:

Code

```
seats                                   =                                   {



}
```

The lines between this blocks curly brackets are the perfect place for the line that finally causes the zombies to rise and infest Your vehicle. You can add as many of those lines as You like; each one will call for one zombie each. Here it is:

Code

```
{ group = 17, transf = transf.rotZYXTransl(transf.degToRad(-10, -12, 0.0), vec3.new(9.8,1,-8
```

Just copy and paste it in there as often as required.

Let us now look at the variables and values that this line takes in order to work:

{ group = p, transf = transf.rotZYXTransl(transf.degToRad(a, b, c), vec3.new(x, y, z), crew = v, standing = w},

p: the sequence number of the part (.msh or .grp) the zombie is attached to. The "children=" block in the .mdl contains all of them; the uppermost one is number 1 and they are simply counted by their order in that file. In most cases, picking the part that contains the seat to be filled will be appropriate; it is also possible to make them roll and tumble along by attaching them to a wheel or spin them dry together with a propeller.

a, b, c: the direction the zombie is facing expressed in degrees about the z, y, x axes. 0, 0, 0 will have them standing upright, facing in the positive direction of the x axis.

x, y, z: the coordinates of the zombies feet. Please note though that these are not absolute coordinates: the reference point is **not** the models origin but the origin of the part p they are attached to. So in order to obtain the coordinates to use here, take the absolute coordinates of the zombies intended position and subtract from them the coordinates of part p. Placing them properly often takes some attempts; the model viewer is a highly useful tool for this and safe to use in auto-reload mode, as no retexturing is involved.

v: a boolean variable. "true" will yield that ill-tempered dude with no interest whatsoever at all in his work that will always loiter around in the vehicle as soon as it is in operation and not used as asset. "false" will result in a passenger seat being defined; the passenger model will be selected by the game based on who is waiting to be transported. Of course, the game will fill the thusly defined passenger seats based on demand; if there is no passenger to be transported, it will be empty.

So the above example requests a sitting passenger attached to part number 17, turned outward by 12°, leaning back by 10° and located 9.8m forward, 1m left and 8.9m downward of part 17s reference point.) Image not found or type u

TPF2 takes some further definitions.

Code

```
{ animation = "sitting", crew = false, forward = true, group = 17, transf = transf.rotZYXTra
```

The additional data is as follows:

Animations define what the zombie is doing.

"sitting" is suitable for passengers loitering on their seats.

"driving" fits those operating a vehicle if sitting low in their seat as in a car.

"driving_upright" is the same but more suited to trucks, trains or aircraft that come with more upright seating.

Crews take one additonal information in TPF2.

Code

```
drivingLicense = "x",
```

equips the zombie with a suitable work attire.

"AIR": a pilot in a grey uniform.

"AIR_OUTDOOR": a pilot with a pilot´s goggle and a scarf.

"BUS": Bus driver

"TRUCK": Truck driver

"TRAM": Light rail driver

"RAIL": Heavy rail driver.

"WATER": Sea Captain

The number of seats you define needs not be related to the actual number of passenger seats called for in the .mdl. If you define a number of seats below the vehicles capacity, the game will just fill the defined seats proportionally to the passenger load and assume all the other passengers to be seated somewhere in the bilge, the coal bunkers or wherever else out of sight. If, on the other hand, you define more seats than required by the capacity, the passengers will take a seat wherever they please and there will be empty seats left with the vehicle filled to capacity., introducing some randomness.

A word of warning: zombies are not well behaved and do not like to sit still. They tend to flail their extremities about and even a solid brick wall can not keep them contained. Do check their seating. If you find zombie parts rammed through the walls, turn them a little inboard about the x axis or just push them over by say 10 or 20cm and let them pester their neighbor instead of kicking your stuff.

Have fun and good luck!